Computer Applications in Experimental Physics

I. Objective

To understand the hardware and software involved in using computers in experimental physics applications.

II. Introduction

One goal of Physics 232 is to present various uses of the computer in physics. Up to now we have been concerned with numerical solutions to mathematical problems. Today we will focus on how computers may be used to acquire and analyze data. We will discuss how computers may be interfaced to experiments through different types of **data buses**, and discuss how software controls the process of data acquisition.

We will then perform a lab to emphasize the key points by writing a program in **QuickBasic** to take data from a Keithley multimeter. You will also use Excel to analyze the data by performing a least squares fit.

III. Exercises

A. Field Trip

Objective:	to be introduced to the hardware and software involved in interfacing a computer to a measuring device
Where to begin:	in room A300
What to do:	listen to a brief talk and take notes
What to turn in to your instructor:	your log book
What to put in log book:	notes from talk

(1) Interfacing Computers: Today we visit a laboratory and hear a brief talk about how they use computers to control experiments. The focus will be on the hardware and software involved in interfacing a computer to laboratory instruments.

(2) Hardware: Currently there exist many industry standards for transferring data from a measuring device to a computer. The sockets on the back of the computer

that accept connections to external devices for the transfer of data are called **data buses** or **data ports**. Two of the most widely used go by the names of **GPIB** (also known as **IEEE-488**) and **RS-232,** though many others standards exist. The table below[1] summarizes the uses of each data bus.

Name	Type	Maximum Data Rate (Mbyte/s)	Use
GPIB (General Purpose Interface Bus)	Parallel	1	May be used to acquire data from a maximum of 15 measuring devices simultaneously by **daisy chaining** connectors
RS-232	Serial	0.002	May be used to acquire data from one measuring device.

(3) Software: In order to acquire data from a measuring device, the computer must run software that controls the flow of data from the device to the computer. Programs for doing this may be written in one of several different languages. A table of different programming languages and their characteristics is given below[2].

Language	Characteristics
Quick BASIC	Easy to learn, but not well structured
FORTRAN	Easy to learn, but not well structured
PASCAL	Easy to learn and the language is highly structured, making code easy to read
C	Highly structured, and many operating systems are written in C
C++	Has all the advantages of C and is **object oriented**
Assembly Language	Fast; some time critical tasks may only be performed with assembly language

(4) C++ Programming (Optional): Dr. Zollner's group writes most of their software for data acquisition in C++. There are good reasons for this: the fact that C++ is

---

[1] J. R. Matey, *Computers in Experimental Physics,* **7**, 130 (1993)

[2] S. Zollner, during a discussion on programming languages.

object oriented reduces the repetition of identical commands and makes the program easier to understand. One way that C++ does this is by defining **classes.**

For example[3], suppose you have your computer interfaced to several different GPIB devices. Because device specifications are different, one would typically have to write separate programs for each device. However, in C++ one may define a **class** for all GPIB devices, which contains all the variables that the computer needs for taking data from GPIB devices. An example of how such a class might look written in C++ is given below:

```
class GPIBdevice{
        int address;
        char name[];
        float getdata(void);
        float data;
        GPIBdevice(address,name)
};
```

For a specific GPIB device, a voltmeter for example, one would specify quantities specific to the device in an **object** called **voltmeter,** as illustrated below:

```
GPIBdevice voltmeter(2,"voltmeter");
```

For this reason, this type of programming is called "object oriented" programming.

B. Taking Your Own Data via a GPIB Port

Objective:	to write a QuickBasic program to take data from a Keithley multimeter through a GPIB port
Where to begin:	in room 82 (Physics 222 Lab), page 27 of Lab #12
What to do:	follow the instructions below
What to turn in to your instructor:	your log book, a copy of your QuickBasic program
What to put in log book:	time you begin/end your work, problems encountered, solutions developed, interesting facts.

(1) Starting QuickBasic: To open the QuickBasic programming environment, type the command **qb <return>** at the **C:\>** prompt. You should see a blue window with the standard menus across the top.

[3] You may find helpful programming examples on the Web site http://femto.ssp.ameslab.gov/os2/gpib/gpib.htm.

(2) Loading and Running Program to Read Multimeter: Parts of what we will do in today's lab you did in Physics 222 during a lab called *The Computer as a Lab Instrument* (Lab #12). In case you do not have a copy of this lab with you, there should be a desk copy of the lab description at your lab table. For an overview of QuickBasic, see pages 11-20 of Lab #12.

Turn to page 27 in Lab #12, follow the instruction there, and record necessary information in your log book. A simple version of a QuickBasic program designed to read data from the multimeter is given on page 28 along with an explanation of the code.

The program on page 28 is stored in **C:\CLI\SAMPLE1.BAS**, which you may load using the command **Open Program** in the **File** menu. To run the program, select the **Run** menu and the command **Start.** You should find that the resistance reading from the multimeter is printed out on the screen without a decimal point.

(3) Modifying the Program: Modify the program to perform the following operations:

- Read resistance values from the multimeter until the key **Q** is pressed. Note that because the decimal point is not read, you will have to make the program divide the resistance reading by an appropriate power of ten.
- Convert the resistance values into Celsius temperature values. Page 31 of Lab#12 shows that the conversion formula for resistance values to Celsius temperatures is given by

$$T(C) = \left(\frac{100 - 0}{138.5 - 100.0} \right) (R - 100)$$

  Show this derivation in your log book.
- Write the time and the temperature into a file called **data.dat**

If you have questions about this, see Appendix 1 for an example.

(4) Taking Data: When your program is running properly, obtain a cup of ice. Begin your program with the resistance thermometer at room temperature. With the program running, place the thermometer in a cup of ice. Continue to take data until the temperature stabilizes.

C. Analyzing Your Data

Objective:	to perform a least squares fit using Excel
Where to begin:	in Excel
What to do:	follow the instructions below
What to turn in to your instructor:	your log book, a copy of your Excel graph
What to put in log book:	time you begin/end your work, problems encountered, solutions developed, interesting facts.

(1) Experimental Model: According to Newton's Law of Cooling, an object looses heat at a rate proportional to the difference between its temperature and the temperature of the reservoir it is in contact with. If the reservoir temperature is T_R and the initial temperature is T_0, then the temperature at any time t is given by

$$T = (T_0 - T_R)e^{-Ct} + T_R$$

We will fit the data taken in the experiment in part B to the above formula using the method of least squares.

(2) Least Squares Fitting: In the method of least squares, one tries to minimize the quantity Δ defined as

$$D \equiv \sum_{i=1}^{N} D_i$$

where $D_i \equiv (T_i^{exp} - T_i^{theor})^2$, T_i^{exp} and T_i^{theor} represent respectively the experimental and theoretical values of the measured quantity at data point i, and the summation is taken over all N data points. Therefore, Δ measures how much the theoretical curve deviates from the experimental curve over the entire data range. Minimizing Δ is one criterion for selecting good fits to experimental data.

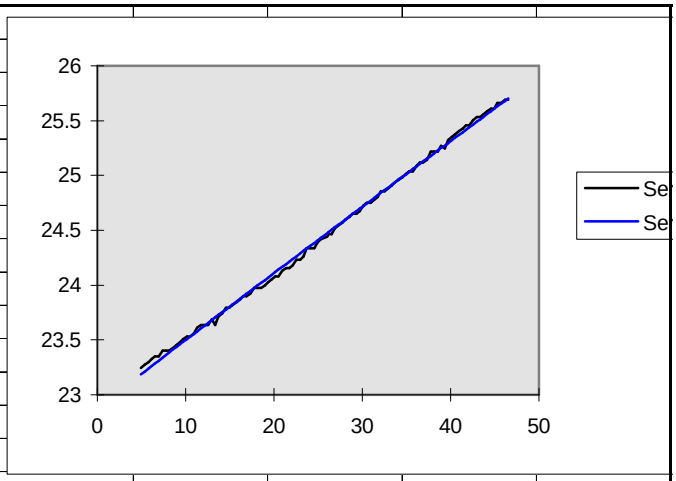
(3) Importing Data into Excel: Import the data you took in part B into your Excel spreadsheet by using the **Open** command under the **File** menu. Insert a few extra rows at the top of your data.

Set up your spread sheet as indicated below, with cells at the top containing parameters T_0, T_R, and C. The value of T_0 is determined by your initial temperature value, and that of T_R may be assumed to be 0 C for the ice water bath. The parameter C is as yet unknown, so make a guess.

Create two additional columns, one for the theoretical temperature T_i^{theor} value expected from Newton's Law of Cooling above (this must reference the cells

containing T_0, T_R, and C), and one for the square of the difference $D_i \equiv (T_i^{exp} - T_i^{theor})^2$. Next, create a cell at the top of the spread sheet called **sumdelta** that contains the sum $D \equiv \sum_{i=1}^{N} D_i$, and set up a graph of T_i^{exp} and T_i^{theor} as shown.

Ti	Tr	C	sumdelta
22.88369	91.60374	0.0009	0.027557
time (s)	Texp (C)	Ttheor (C)	delta
4.941406	23.24675	23.1886	0.003382
5.332031	23.27272	23.21264	0.003609
5.761719	23.2987	23.23908	0.003554
6.152344	23.32468	23.26311	0.003791
6.53125	23.35064	23.28641	0.004125
6.921875	23.35064	23.31042	0.001618
7.359375	23.4026	23.3373	0.004264
7.742188	23.4026	23.36082	0.001746
8.132813	23.4026	23.3848	0.000317
8.570313	23.42856	23.41166	0.000286
8.953125	23.45454	23.43514	0.000376
9.332031	23.48052	23.45838	0.00049



(4) Performing the Least Squares Fit: Suppose we wish now to minimize the variable **sumdelta** by varying the parameter **C**. This is a one parameter least squares fit.

To perform the minimization of **sumdelta**, we will use a function called **Solver.** Select the **Tools** menu and select the **Solver** function. A box will open containing several prompts. Locate the prompt **Select target cell** and specify the cell address of **sumdelta,** the quantity we desire to minimize**.** Next locate the prompt **Equal to** and select the minimization button **min**. Finally locate the prompt **By changing cells** and select the cell location of the parameters to be adjusted, in this case **C.**

After the above selections have been made, begin the minimization by clicking the button **Solve.** You should see the iterations begin. After several iterations, Excel will modify the graph of T_i^{theor} for the new value of C that minimized the sum of the squares. Record this value of C in your log book print out a copy of the graph showing the data and the fit to the data.

(5) Playing with the Fit: Now try fitting with two and three parameters. Are the results more or less consistent with what you expect? Report and discuss your findings in your log book.

IV. Appendix 1: QuickBasic Program for Reading Keithley Multimeter

Listed below is one way to write a program that will read resistance values from a Keithley Model 177 Multimeter. The program writes the time and temperature to the screen and to the file **C:\CLI\DATA.DAT**. The program continues to take data until the **Q** key is pressed.

```
7000  'Reading a KEITHLEY model 177 multimeter via the IEEE-488 interface
7005  CLS
7010  OPEN "GPIB0" FOR OUTPUT AS #1          'PREPARE GPI BOARD #0 FOR OUTPUT TO METER
7020  OPEN "GPIB0" FOR INPUT AS #2           'PREPARE GPI BOARD #0 FOR INPUT FROM METER
7025  OPEN "C:\CLI\DATA.DAT" FOR OUTPUT AS #3   'OPENS DATA FILE
7026  INIT = TIMER                           'GETS INITIAL TIME IN SEC
7030  PRINT #1, "ENTER 12#8"                 'TELL METER AT ADDRESS 12 TO PRESENT A READING OF 8 CHARACTERS
7040  INPUT #2, R$                           'ACCEPT READING INTO MEMORY RESERVED FOR R$
7050  R = VAL(R$) / 100                      'CONVERT CHARACTERS INTO A NUMBER
7055  C = 100 / 38.5 * (R - 100)             'CALCULATES CELCIUS TEMPERATURE
7060  PRINT TIMER - INIT, C                  'PRINTS TIME(SEC) AND TEMP(C) TO SCREEN
7063  PRINT #3, TIMER - INIT, C              'PRINTS TO FILE C:\CLI\DATA.DAT
7065  IF INKEY$ = "Q" THEN GOTO 7080         'TERMINATES IF Q KEY PRESSED
7070  GOTO 7030                              'LOOPS BACK FOR ANOTHER DATA POINT
7080  END
```